DOCUMENT RESUME

ED 459 819 IR 058 355

AUTHOR	Chau, Michael; Zeng, Daniel; Chen, Hsinchun
TITLE	Personalized Spiders for Web Search and Analysis.
SPONS AGENCY	National Science Foundation, Arlington, VA. Directorate for Computer and Information Science and Engineering.
PUB DATE	2001-06-00
NOTE	11p.; In: Proceedings of the ACM/IEEE-CS Joint Conference on Digital Libraries (1st, Roanoke, Virginia, June 24-28, 2001). For entire proceedings, see IR 058 348. Also supported by NSF Computation and Social Systems Program. Figures may not reproduce well.
CONTRACT	IIS-9817473; IIS-9800686
AVAILABLE FROM	Association for Computing Machinery, 1515 Broadway, New York NY 10036. Tel: 800-342-6626 (Toll Free); Tel: 212-626-0500; e-mail: acmhelp@acm.org. For full text: http://www1.acm.org/pubs/contents/proceedings/dl/379437/.
PUB TYPE	Reports - Research (143) -- Speeches/Meeting Papers (150)
EDRS PRICE	MF01/PC01 Plus Postage.
DESCRIPTORS	*Information Retrieval; Information Systems; Internet; *Online Searching; Search Strategies; *User Needs (Information); *World Wide Web
IDENTIFIERS	Browsing; Search Engines; Spiders

ABSTRACT

Searching for useful information on the World Wide Web has become increasingly difficult. While Internet search engines have been helping people to search on the Web, low recall rate and outdated indexes have become more and more problematic as the Web grows. In addition, search tools usually present to the user only a list of search results, failing to provide further personalized analysis which could help users identify useful information and comprehend these results. To alleviate these problems, the authors propose a client-based architecture that incorporates noun phrasing and self-organizing map techniques. Two systems, namely CI Spider and Meta Spider, have been built based on this architecture. User evaluation studies have been conducted and the findings suggest that the proposed architecture can effectively facilitate Web search and analysis. (Contains 23 references.) (Author)

Reproductions supplied by EDRS are the best that can be made
from the original document.

ED 459 819

Personalized Spiders for Web Search and Analysis

By: Michael Chau, Daniel Zeng & Hsinchun Chen

PERMISSION TO REPRODUCE AND DISSEMINATE THIS MATERIAL HAS BEEN GRANTED BY

D. Cotton

TO THE EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)

1

U.S. DEPARTMENT OF EDUCATION
Office of Educational Research and Improvement
EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)

☐ This document has been reproduced as received from the person or organization originating it.

☐ Minor changes have been made to improve reproduction quality.

• Points of view or opinions stated in this document do not necessarily represent official OERI position or policy.

IR058355

2

BEST COPY AVAILABLE

Personalized Spiders for Web Search and Analysis

Michael Chau
Dept of Management Info. Sys.
The University of Arizona
Tucson, Arizona 85721, USA
1-520-626-9239
mchau@bpa.arizona.edu

Daniel Zeng
Dept of Management Info. Sys.
The University of Arizona
Tucson, Arizona 85721, USA
1-520-621-4614
zeng@bpa.arizona.edu

Hsinchun Chen
Dept of Management Info. Sys.
The University of Arizona
Tucson, Arizona 85721, USA
1-520-621-2748
hchen@bpa.arizona.edu

ABSTRACT
Searching for useful information on the World Wide Web has become increasingly difficult. While Internet search engines have been helping people to search on the web, low recall rate and outdated indexes have become more and more problematic as the web grows. In addition, search tools usually present to the user only a list of search results, failing to provide further personalized analysis which could help users identify useful information and comprehend these results. To alleviate these problems, we propose a client-based architecture that incorporates noun phrasing and self-organizing map techniques. Two systems, namely CI Spider and Meta Spider, have been built based on this architecture. User evaluation studies have been conducted and the findings suggest that the proposed architecture can effectively facilitate web search and analysis.

Categories and Subject Descriptors
H.3.3 **[Information Storage and Retrieval]**: Information Search and Retrieval – *clustering, information filtering, search process.*

General Terms
Design, Experimentation.

Keywords
Information retrieval, Internet spider, Internet searching and browsing, noun-phrasing, self-organizing map, personalization, user evaluation.

1. INTRODUCTION
The World Wide Web has become the biggest digital library available, with more than 1 billion unique indexable web pages [9]. However, it has become increasingly difficult to search for useful information on it, due to its dynamic, unstructured nature and its fast growth rate. Although development of web search and analysis tools such as search engines has alleviated the problem to a great extent, exponential growth of the web is making it impossible to collect and index all the web pages and refresh the index frequently enough to keep it up-to-date. Most search engines present search results to users that are incomplete and outdated, usually leaving users confused and frustrated.

A second problem that Internet users encounter is the difficulty in searching information on a particular website, e.g., looking for information related to a certain topic in the website www.phoenix.com. Among the popular commercial search engines, only a few offer the search option to limit a search session to a specified website. Because most search engines only index a certain portion of each website, the recall rate of these searches is very low, and sometimes even no documents are returned. Although most large websites nowadays have their built-in internal search engines, these engines index the information based on different schemes and policies and users may have difficulty in uncovering useful information. In addition, most of the websites on the Internet are small sites that do not have an internal search feature.

A third problem is the poor retrieval rate when only a single search engine is used. It has been estimated that none of the search engines available indexes more than 16% of the total web that could be indexed [12]. Even worse, each search engine maintains its own searching and ranking algorithm as well as query formation and freshness standard. Unless the different features of each search engine are known, searches will be inefficient and ineffective. From the user's point of view, dealing with an array of different interfaces and understanding the idiosyncrasies of each search engine is too burdensome. The development of meta-search engines has alleviated this problem. However, how the different results are combined and presented to the user greatly affects the effectiveness of these tools.

In addition, given the huge number of daily hits, most search engines are not able to provide enough computational power to satisfy each user's information need. Analysis of search results, such as verifying that the web pages retrieved still exist or clustering of web pages into different categories, are not available in most search engines. Search results are usually presented in a ranked list fashion; users cannot get a whole picture of what the web pages are about until they click on every page and read the contents. This can be time-consuming and frustrating in a dynamic, fast-changing electronic information environment.

In order to alleviate the above problems, we propose a personalized and integrated approach to web search. In this paper, we present a client-side web search tool that applies various artificial intelligence techniques. We believe that a search tool that is more customizable would help users locate useful

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
JCDL'01, June 24-28, 2001, Roanoke, Virginia, USA.
Copyright 2001 ACM 1-58113-345-6/01/0006...$5.00.

information on the web more effectively. The client-based architecture also allows for greater computation power and resources to provide better searching and analysis performance. We have conducted two experiments to evaluate the performance of different prototypes built according to this architecture.

2. RELATED WORK

In order to address the information overload problem on the web, research has been conducted in developing techniques and tools to analyze, categorize and visualize large collections of web pages, among other text documents. A variety of tools have been developed to assist searching, gathering, monitoring and analyzing information on the Internet.

2.1 Web Search Engines and Spiders

Many different search engines are available on the Internet. Each has its own characteristics and employs its preferred algorithm in indexing, ranking and visualizing web documents. For example, AltaVista (www.altavista.com) and Google (www.google.com) allow users to submit queries and present web pages in a ranked order, while Yahoo! (www.yahoo.com) groups websites into categories, creating a hierarchical directory of a subset of the Internet.

Another type of search engine is comprised of meta-search engines, such as MetaCrawler (www.metacrawler.com) and Dogpile (www.dogpile.com). These search engines connect to multiple search engines and integrate the results returned. As each search engine covers different portion of the Internet, meta-search engines are useful when the user needs to get as much of the Internet as possible. There are also special-purpose topic-specific search engines [4]. For example, BuildingOnline (www.buildlingonline.com) specializes in searching in the building industry domain on the web, and LawCrawler (www.lawcrawler.com) specializes in searching for legal information on the Internet.

Internet spiders (a.k.a. crawlers), have been used as the main program in the backend of most search engines. These are programs that collect Internet pages and explore outgoing links in each page to continue the process. Examples include the World Wide Web Worm [16], the Harvest Information Discovery and Access System [1], and the PageRank-based Crawler [5].

In recent years, many client-side web spiders have been developed. Because the software runs on the client machine, more CPU time and memory can be allocated to the search process and more functionalities are possible. Also, these tools allow users to have more control and personalization options during the search process. For example, Blue Squirrel's WebSeeker (www.bluesquirrel.com) and Copernic 2000 (www.copernic.com) connect with different search engines, monitor web pages for any changes, and schedule automatic search. Focused Crawler [2] locates web pages relevant to a pre-defined set of topics based on example pages provided by the user. In addition, it also analyzes the link structures among the web pages collected.

2.2 Monitoring and Filtering

Because of the fast changing nature of the Internet, different tools have been developed to monitor websites for changes and filter out unwanted information. Push Technology is one of the emerging technologies in this area. The user first needs to specify some areas of interest. The tool will then automatically push related information to the user. Ewatch (www.ewatch.com) is one such example. It monitors information not only from web pages but also from Internet Usenet groups, electronic mailing lists, discussion areas and bulletin boards to look for changes and alert the user.

Another popular technique used for monitoring and filtering employs a software agent, or intelligent agent [15]. Personalized agents can monitor websites and filter information according to particular user needs. Machine learning algorithms, such as an artificial neural network, are usually implemented in agents to learn the user's preferences.

2.3 Indexing and Categorization

There have been many studies in textual information analysis of information retrieval and natural language processing. In order to retrieve documents based on given concepts, the documents have to be indexed. Automatic indexing algorithms have been used widely to extract key concepts from textual data. It having been shown that automatic indexing is as effective as human indexing [18], many proven techniques have been developed. Linguistics approaches such as noun phrasing also have been applied to perform indexing for phrases rather than just words [21]. These techniques are useful in extracting meaningful terms from text documents not only for document retrieval but also for further analysis.

Another type of analysis tool is categorization. These tools allow a user to classify documents into different categories. Some categorization tools facilitate the human categorization process by simply providing a user-friendly interface. Tools that are more powerful categorize documents automatically, allowing users to quickly identify the key topics involved in a large collection of documents [e.g., 8, 17, 23].

In document clustering, there are in general two approaches. In the first approach, documents are categorized based on individual document attributes. An attribute might be the query term's frequency in each document [7, 22]. NorthernLight, a commercial search engine, is another example of this approach. The retrieved documents are organized based on the size, source, topic or author of each document. Other examples include Envision [6] and GRIDL [19].

In the second approach, documents are classified based on inter-document similarities. This approach usually includes some kind of machine learning algorithms. For example, the Self-Organizing Map (SOM) approach classifies documents into different categories which are defined during the process, using neural network algorithm [10]. Based on this algorithm, the SOM technique automatically categorizes documents into different regions based on the similarity of the documents. It produces a data map consisting of different regions, where each region contains similar documents. Regions that are similar are located close to each other. Several systems utilizing this technique have been built [3, 11, 14].

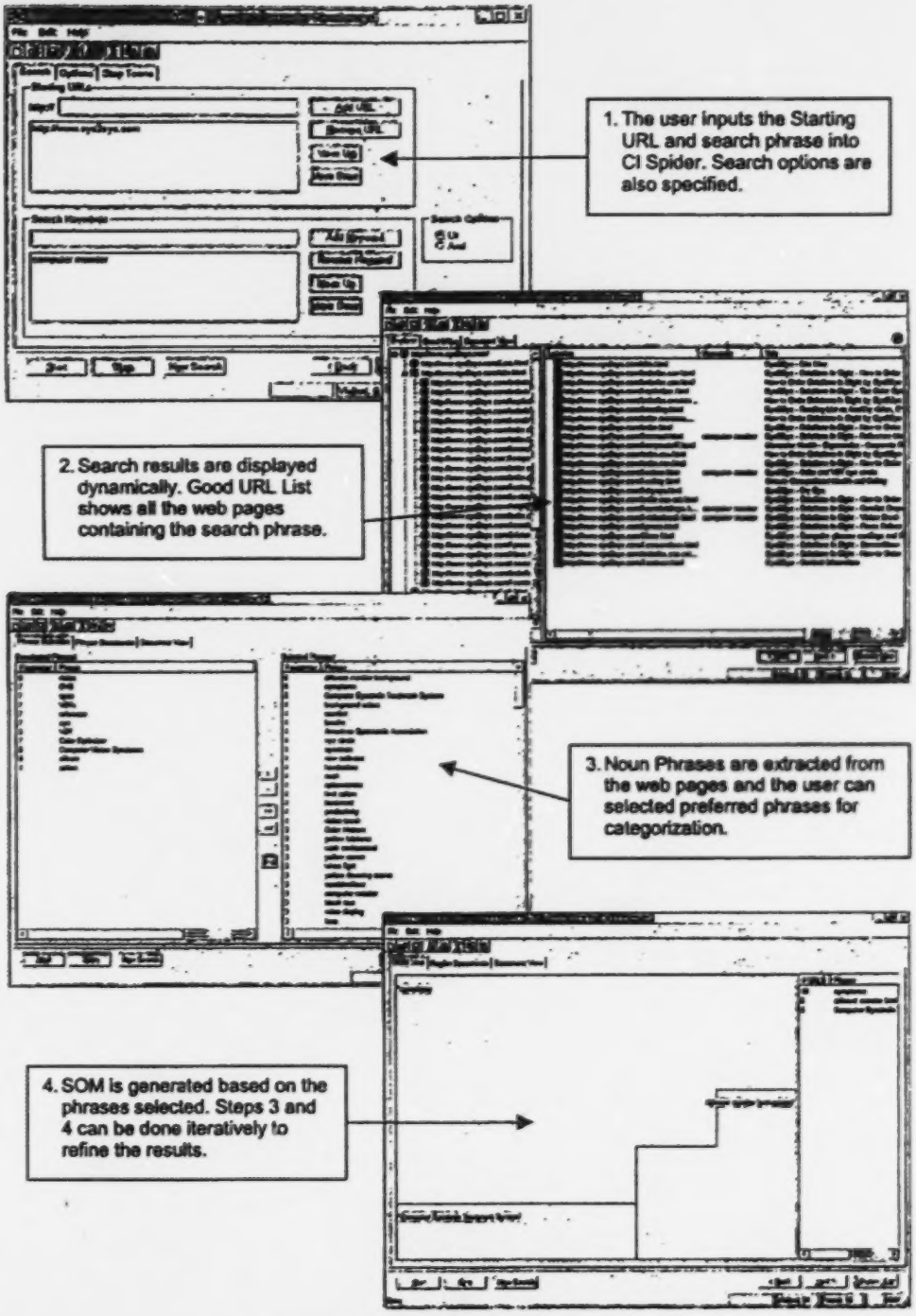
1. The user inputs the Starting URL and search phrase into CI Spider. Search options are also specified.
2. Search results are displayed dynamically. Good URL List shows all the web pages containing the search phrase.
3. Noun Phrases are extracted from the web pages and the user can selected preferred phrases for categorization.
4. SOM is generated based on the phrases selected. Steps 3 and 4 can be done iteratively to refine the results.

Figure 1. Example of a User Session with CI Spider

BEST COPY AVAILABLE

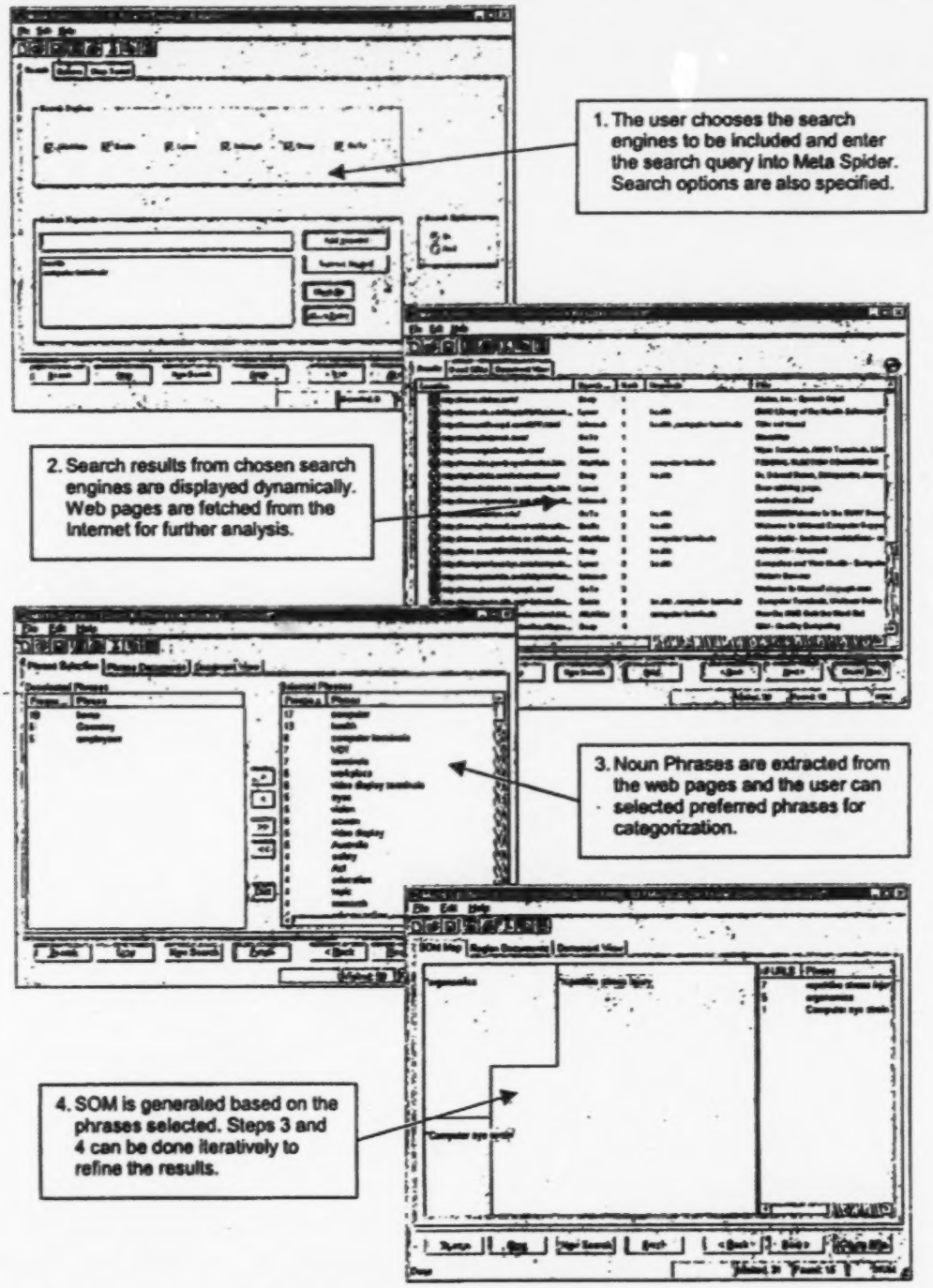
1. The user chooses the search engines to be included and enter the search query into Meta Spider. Search options are also specified.
2. Search results from chosen search engines are displayed dynamically. Web pages are fetched from the Internet for further analysis.
3. Noun Phrases are extracted from the web pages and the user can selected preferred phrases for categorization.
4. SOM is generated based on the phrases selected. Steps 3 and 4 can be done iteratively to refine the results.

Figure 2. Example of a User Session with Meta Spider

BEST COPY AVAILABLE

3. SYSTEM DESIGN

Two different prototypes based on the proposed architecture have been built. Competitive Intelligence Spider, or CI Spider, collects web pages on a real-time basis from websites specified by the user and performs indexing and categorization analysis on them, to provide the user with a comprehensive view of the websites of interest. A sample user session with CI Spider is shown in Figure 1. The second tool, Meta Spider, has similar functionalities as the CI Spider, but instead of performing breadth-first search on a particular website, connects to different search engines on the Internet and integrates the results. A sample user session with Meta Spider is shown in Figure 2.

The architecture of CI Spider and Meta Spider is shown in Figure 3. There are 4 main components, namely (1) User Interface, (2) Internet Spiders, (3) Noun Phraser, and (4) Self-Organizing Map (SOM). These components work together as a unit to perform web search and analysis.

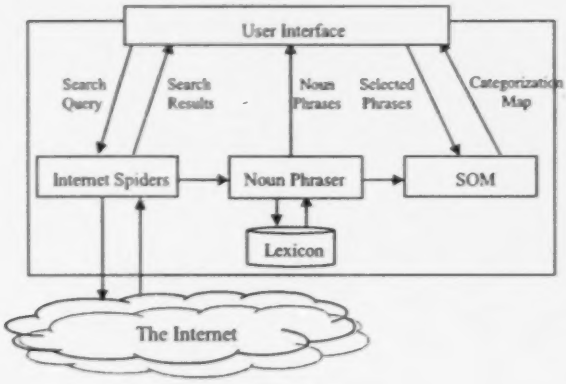

Figure 3. System Architecture

3.1 Internet Spiders

In CI Spider, the Internet Spiders are Java spiders that start from the URLs specified by the user and follow the outgoing links to search for the given keywords, until the number of web pages collected reaches a user-specified target. The spiders run in multi-thread such that the fetching process will not be affected by slow server response time. Robots exclusion protocol is also implemented such that the spiders will not access sites where the web master has placed a text file in a host or a meta-tag in a web page, indicating that robots are not welcome to these sites.

In the case of Meta Spider, the Internet Spiders first send the search queries to the search engines chosen. After the results are obtained, the Internet Spiders attempt to fetch every result page. Deadlinks and pages which do not contain the search keyword are discarded.

Whenever a page is collected during the search, the link to that page is displayed dynamically. The user can click on any link displayed and read its full content without having to wait for the whole search to be completed. The user can also switch to the Good URL List to browse only the pages that contain the search keyword. When the number of web pages collected meets the amount specified by the user, the spiders will stop and the results will be sent to the Noun Phraser for analysis.

3.2 Noun Phraser

The Arizona Noun Phraser developed at the University of Arizona is the indexing tool used to index the key phrases that appear in each document collected from the Internet by the Internet Spiders. It extracts all the noun phrases from each document based on part-of-speech tagging and linguistic rules [21]. The Arizona Noun Phraser has three components. The tokenizer takes web pages as text input and creates output that conforms to the UPenn Treebank word tokenization rules by separating all punctuation and symbols from text without interfering with textual content. The tagger module assigns a part-of-speech to every word in the document. The last module, called the phrase generation module, converts the words and associated part-of-speech tags into noun phrases by matching tag patterns to a noun phrase pattern given by linguistic rules. Readers are referred to [21] for more detailed discussion. The frequency of every phrase is recorded and sent to the User Interface. The user can view the document frequency of each phrase and link to the documents containing that phrase. After all documents are indexed, the data are aggregated and sent to the Self-Organizing Map for categorization.

3.3 Self-Organizing Map (SOM)

In order to give users an overview of the set of documents collected, the Kohonen SOM employs an artificial neural network algorithm to automatically cluster the web pages collected into different regions on a 2-D map [10]. Each document is represented as an input vector of keywords and a two-dimensional grid of output nodes is created. After the network is trained, the documents are submitted to the network and clustered into different regions. Each region is labeled by the phrase which is the key concept that most accurately represents the cluster of documents in that region. More important concepts occupy larger regions, and similar concepts are grouped in a neighborhood [13]. The map is displayed through the User Interface and the user can view the documents in each region by clicking on it.

3.4 Personalization Features

Because both CI Spider and Meta Spider have been designed for personalized web search and analysis, a user has been given more control during the search process.

In the Options Panel, the user can specify how the search is to be performed. This is similar to the "Advanced Search" feature of some commercial search engines. The user can specify number of web pages to be retrieved, domains (e.g. .gov, .edu or .com) to be included in the search results, number of Internet Spiders to be used, and so on. In CI Spider, the user can also choose either Breadth-First Search or Best-First Search to be the algorithm used by the Internet Spiders.

The SOM also is highly customizable in the sense that the user can select and deselect phrases for inclusion in the analysis and produce a new map at any time. If the user is not satisfied with the map produced, he can always go back to the previous step to discard some phrases that are irrelevant or too general and

generate a new map within seconds. The systems also let each user store a personalized "dictionary" which contains the terms that the user does not want to be included in the results of the Arizona Noun Phraser and the SOM.

Another important functionality incorporated in the system is the Save function. The user can save a completed search session and open it at a later time. This feature allows the user to perform a web search and review it in the future. This also helps users who want to monitor web pages on a particular topic or website.

4. EVALUATION METHODOLOGIES

Two separate experiments have been conducted to evaluate CI Spider and Meta Spider. Because we designed the two spider systems to facilitate both document retrieval and document categorization tasks, traditional evaluation methodologies would not have been appropriate. These methodologies treat document retrieval and document categorization separately. In our experiments, the experimental task was therefore so designed as to permit evaluation of the performance of a combination of their functionalities in identifying the major themes related to a certain topic being searched.

4.1 Evaluation of CI Spider

In our experiment, CI Spider was compared with the usual methods that Internet users use to search for information on the Internet. General users usually use popular commercial search engines to collect data on the Internet, or they simply explore the Internet manually. Therefore, these two search methods were compared with the CI Spider. The first method evaluated was Lycos, chosen because it is one of the few popular search engines that offer the functionality to search for a certain keyword in a given web domain. The second method was "within-site" browsing and searching. In this method the subject was allowed to freely explore the contents in the given website using an Internet browser. When using CI Spider, the subject was allowed to use all the components including Noun Phraser and SOM.

Each subject first tried to locate the pages containing the given topic within the given web host using the different search methods described above. The subject was required to comprehend the contents of all the web pages relevant to that keyword, and to summarize the findings as a number of themes. In our experiment, a theme was defined as "a short phrase which describes a certain topic." Phrases like "success of the 9840 tape drive in the market" and "business transformation services" are examples of themes in our experiment. By examining the themes that the subjects came up with using different search methods, we were able to evaluate how effectively and efficiently each method helped a user locate a collection of documents and gain a general understanding of the response to a given search query on a certain website. Websites with different sizes, ranging from small sites such as www.eye2eye.com to large sites such as www.ibm.com were chosen for the experiments.

Six search queries were designed for the experiment, based on suggestions given by professionals working in the field of competitive intelligence. For example, one of our search tasks was to locate and summarize the information related to "merger" on the website of a company called Phoenix Technologies (www.phoenix.com). Two pilot studies were conducted in order to refine the search tasks and experiment design. During the real experiment, thirty subjects, mostly information systems management students, were recruited and each subject was required to perform three out of the six different searches using the three different search methods. At the beginning of each experiment session, the subject was trained in using these search methods. Each subject performed at least one complete search session for each of the 3 search methods until he felt comfortable with each method. Rotation was applied such that the order of search methods and search tasks tested would not bias our results.

4.2 Evaluation of Meta Spider

Meta Spider was compared with MetaCrawler and NorthernLight. MetaCrawler (www.metacrawler.com) is a renowned, popular meta-search engine and has been recognized for its adaptability, portability and scalability [20]. NorthernLight (www.northernlight.com), being one of the largest search engines on the web, provides clustering functionality to classify search results into different categories. When using Meta Spider, the subject was allowed to use all the components including Noun Phraser and SOM.

Each subject was required to use the different search tools to collect information related to the given topic. As in the CI Spider experiment, each subject was required to summarize the web pages collected as a number of themes. The search topics were chosen from TREC 6 topics. Because the TREC topics were not especially designed for web document retrieval, care was taken to make sure each search topic was valid and retrievable on the Internet. Thirty undergraduate students from an MIS class at The University of Arizona were recruited to undertake the experiment. Training and rotation similar to those used in the CI Spider experiment were applied.

5. EXPERIMENT RESULTS AND DISCUSSION

Two graduate students majoring in library science were recruited as experts for each experiment. They employed the different search methods and tools being evaluated and came up with a comprehensive set of themes for each search task. Their results were then aggregated to form the basis for evaluation. Precision and recall rates for themes were used to measure the effectiveness of each search method.

The time spent for each experiment, including the system response time and the user browsing time, was recorded in order to evaluate the efficiency of the 3 search methods in each experiment. During the studies, we encouraged our subjects to tell us about the search method used and their comments were recorded. Finally, each subject filled out a questionnaire to record further comments about the 3 different methods.

5.1 Experiment Results of CI Spider

The quantitative results of the CI Spider experiment are summarized in Table 1. Four main variables for each subject have been computed for comparison: precision, recall, time, and ease of use. Precision rate and recall rate were calculated as follows:

$$precision = \frac{number\ of\ correct\ themes\ identified\ by\ the\ subject}{number\ of\ all\ themes\ identified\ by\ the\ subject}$$

$$recall = \frac{number\ of\ correct\ themes\ identified\ by\ the\ subject}{number\ of\ all\ themes\ identified\ by\ the\ expert\ judges}$$

The time recorded was the total duration of the search task, including both response time of the system and the browsing time of the subject. Usability was calculated based on subjects' responses to the question "How easy/difficult is it to locate useful information using [that search method]?" Subjects were required to choose a level from a scale of 1 to 5, with 1 being the most difficult and 5 being the easiest.

In order to see whether the differences between the values were statistically significant, *t*-tests were performed on the experimental data. The results are summarized in Table 2. As can be seen, the precision and recall rates for CI Spider both were significantly higher than those of Lycos at a 5% significant level. CI Spider also was given a statistically higher value than Lycos and within-site browsing and searching in usability.

Table 1: Experiment results of CI Spider

		CI Spider	Lycos	Within-Site Browsing/ Searching
Precision:	Mean	**0.708**	0.477	0.576
	Variance	0.120	0.197	0.150
Recall:	Mean	**0.273**	0.163	0.239
	Variance	0.027	0.026	0.033
Time(min):	Mean	10.02	9.23	**8.60**
	Variance	11.86	44.82	36.94
Usability*:	Mean	**3.97**	3.33	3.23
	Variance	1.34	1.13	1.29

*Based on a scale of 1 to 5, where 1 being the most difficult to use and 5 being the easiest.

Table 2: *t*-test results of CI Spider Experiment

	CI Spider vs Lycos	CI Spider vs Within-Site B/S	Lycos vs Within-Site B/S
Precision	***0.029**	0.169	0.365
Recall	***0.012**	0.459	0.087
Time	0.563	0.255	0.688
Usability	***0.031**	***0.016**	0.126

* The mean difference is significant at the 0.05 level.

5.2 Experiment Results of Meta Spider

Three variables, namely precision, recall, and time, have been computed for comparison in the Meta Spider experiment and the results are summarized in Table 3. The *t*-test results are summarized in Table 4. In terms of precision, Meta Spider performed better than MetaCrawler and NorthernLight, and the difference with NorthernLight was statistically significant. For recall rate, Meta Spider was comparable to MetaCrawler and better than NorthernLight.

Table 3: Experiment results of Meta Spider

		Meta Spider	Meta-Crawler	Northern-Light
Precision:	Mean	**0.815**	0.697	0.561
	Variance	0.281	0.315	0.402
Recall:	Mean	0.308	**0.331**	0.203
	Variance	0.331	0.291	0.181
Time(min):	Mean	**10.93**	11.13	11.00
	Variance	4.04	4.72	5.23

Table 4: *t*-test results of Meta Spider Experiment

	Meta Spider vs Meta-Crawler	Meta Spider vs Northern-Light	Meta-Crawler vs Northern-Light
Precision	0.540	***0.013**	0.360
Recall	1.000	0.304	0.139
Time	1.000	1.000	1.000

* The mean difference is significant at the 0.05 level.

5.3 Strength and Weakness Analysis

5.3.1 Precision and Recall

The *t*-test results show that CI Spider performed statistically better in both precision and recall than Lycos, and Meta Spider performed better than NorthernLight in precision. In terms of precision, we suggest that the main reason for the high precision rate of CI Spider and Meta Spider is their ability to fetch and verify the content of each web page in real time. That means our Spiders can ensure that every page shown to the user contains the keyword being searched. On the other hand, we found that indexes in Lycos and NorthernLight, like most other search engines, were often outdated. A number of URLs returned by these two search engines were irrelevant or dead links, resulting in low precision. Subjects also reported that in some cases two or more URLs returned by Lycos pointed to the same page, which led to wasted time verifying the validity of each page.

The high recall rate of CI Spider is mainly attributable to the exhaustive searching nature of the spiders. Lycos has the lowest recall rate because, like most other commercial search engines, it samples only a number of web pages in each website, thereby missing other pages that contain the keyword. For within-site browsing and searching, a user is more likely to miss some important pages because the process is mentally exhausting.

5.3.2 Display and Analysis of Web Pages

In the CI Spider study, subjects believed it was easier to find useful information using CI Spider (with a score of 3.97/5.00) than using Lycos domain search (3.33) or manual within-site browsing and searching (3.23). Three main reasons may account for this. The first is the high precision and recall discussed above. The high quality of data saved users considerable time and mental effort. Second, the intuitive and useful interface design helped subjects locate information they needed more easily. Third, the analysis tools helped subjects form an overview of all the relevant web pages collected. The Arizona Noun Phraser allowed subjects to narrow and refine their searches as well as provided a list of key phrases that represented the collection. The Self-Organizing Map generated a

2-D map display on which subjects could click to view the documents related to a particular theme of the collection.

In our post-test questionnaires in the CI Spider experiment, we found that 77% of subjects found the Good URL List useful for their analyses, while 40% of subjects found either the Noun Phraser or the SOM useful. This suggests that while many subjects preferred traditional search result list, a significant portion of subjects were able to gain from the use of advanced analysis tools. Similar results were obtained in the Meta Spider experiment, in which 77% of subjects found the list display useful and 45% found either the Noun Phraser or the SOM useful.

5.3.3 Speed

The *t*-test results demonstrated that the three search methods in each experiment did not differ significantly in time requirements. As discussed in the previous section, the time used for comparison is total searching time and browsing time. Real-time indexing and fetching time, which usually takes more than 3 minutes, also was included in the total time for CI Spider and Meta Spider. Therefore, we anticipate that the two Spiders can let users spend less time and effort in the whole search process, because the users only need to browse the verified results.

6. CONCLUSION

The results of the two studies are encouraging. They indicate that the use of CI Spider and Meta Spider can potentially facilitate the web searching process for Internet users with different needs by using a personalized approach. The results also demonstrated that powerful AI techniques such as noun phrasing and SOM can be processed on the user's personal computer to perform further analysis on web search results, which allows the user to understand the search topic more correctly and more completely. We believe that many other powerful techniques can possibly be implemented on client-side search tools to improve efficiency and effectiveness in web search as well as other information retrieval applications.

7. ACKNOWLEDGMENTS

We would like to express our gratitude to the following agencies for supporting this project:

- NSF Digital Library Initiative-2, "High-performance Digital Library Systems: From Information Retrieval to Knowledge Management", IIS-9817473, April 1999-March 2002.
- NSF/CISE/CSS, "An Intelligent CSCW Workbench: Personalized Analysis and Visualization", IIS-9800696, June 1998-June 2001.

We would also like to thank all the members of the Artificial Intelligence Lab at the University of Arizona who have contributed in developing the two search tools, in particular Wojciech Wyzga, Harry Li, Andy Clements, Ye Fang, David Hendriawan, Hadi Bunnalim, Ming Yin, Haiyan Fan, Bill Oliver and Esther Chou.

8. REFERENCES

[1] Bowman, C., Danzig, P., Manber, U. and Schwartz, F. Scalable Internet Resource Discovery: Research Problems and Approaches, Communications of the ACM 37(8): 98-107 (1994).

[2] Chakrabarti, S., van der Berg, M. and Dom, B. Focused Crawling: A New Approach to Topic-Specific Web Resource Discovery, in Proceedings of the 8th International World Wide Web Conference (Toronto, Canada, 1999).

[3] Chen, H., Schufels, C. and Orwig, R. Internet Categorization and Search: A Self-Organizing Approach, Journal of Visual Communication and Image Representation, 7(1): 88-102 (1996).

[4] Chignell, M. H., Gwizdka, J. and Bodner, R. C. Discriminating Meta-Search: A Framework for Evaluation. Information Processing and Management, 35 (1999).

[5] Cho, J., Garcia-Molina, H. and Page, L. Efficient Crawling Through URL Ordering, in Proceedings of the 7th World Wide Web Conference (Brisbane, Australia, Apr 1998).

[6] Fox, E., Hix, D., Nowell, L. T., Brueni, D. J., Wake, W. C., Lenwood, S. H. and Rao, D. Users, User Interfaces, and Objects: Envision, A Digital Library. Journal of the American Society for Information Science, 44(8), 480-491 (1993).

[7] Hearst, M. TileBars: Visualization of Term Distribution Information in Full Text Information Access, in Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems (CHI'95), 59-66 (1995).

[8] Hearst, M. and Pedersen, J. Reexamining the Cluster Hypothesis: Scatter/Gather on Retrieval Results, in Proceedings of the 19th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'96), 76-84 (1996).

[9] Inktomi WebMap, http://www.inktomi.com/webmap/

[10] Kohonen, T. Self-Organizing Maps. Springer-Verlag, Berlin, 1995.

[11] Kohonen, T. Exploration of Very Large Databases by Self-Organizing Maps, in Proceedings of the IEEE International Conference on Neural Networks, 1:1-6 (1997).

[12] Lawrence, S. and Giles, C. L. Accessibility of Information on the Web, Nature, 400 (1999), 107-109.

[13] Lin, C., Chen, H. and Nunamaker J. Verifying the Proximity and Size Hypothesis for Self-Organizing Maps. Journal of Management Information Systems, 16(3) (1999-2000), 61-73.

[14] Lin, X., Soergel, D., and Marchionini, G. A Self-organizing Semantic Map for Information Retrieval, in Proceedings of the 14th International ACM SIGIR Conference on Research and Development in Information Retrieval (1991), 262-269.

[15] Maes, P. Agents that Reduce Work and Information Overload. Communications of the ACM, 37(7) (July 1994), 31-40.

[16] McBryan, O. GENVL and WWW: Tools for Taming the Web, in Proceedings of the 1st International World Wide Web Conference (Geneva, Switzerland, Mar 1994).

[17] Rasmussen, E. Clustering Algorithms. In W. B. Frakes and R. Baeza-Yates (eds.) Information Retrieval Data Structures and Algorithms, Prentice Hall, N. J., 1992.

[18] Salton, G. Another look at automatic text-retrieval systems. Communications of the ACM, 29(7) (1986), 648-656.

[19] Schneiderman, B., Feldman, D., Rose, A. and Grau, X. F. Visualizing Digital Library Search Results with Categorical and Hierarchical Axes, in Proceedings of 5th ACM Conference on ACM 2000 Digital Libraries (San Antonio, Texas USA, 2000).

[20] Selberg, E. and Etzioni, O. The MetaCrawler architecture for resource aggregation on the Web. IEEE Expert (1997).

[21] Tolle, K. M. and Chen, H. Comparing Noun Phrasing Techniques for Use with Medical Digital Library Tools. Journal of the American Society for Information Science, 51(4), 352-370 (2000).

[22] Veerasamy, A. and Belkin, N. J., Evaluation of a Tool for Visualization of Information Retrieval Results, in Proceedings of the 19th International ACM SIGIR Conference on Research and Development in Information Reurieval (SIGIR'96), 85-92 (1996).

[23] Zamir, O. and Etzioni, O. Grouper: A Dynamic Clustering Interface to Web Search Results, in Proceedings of the 8th International World Wide Web Conference (Toronto, Canada, May 1999).